**Design Decisions Reflection**

For my final project in CS-330, I was tasked with creating a simple three-dimensional (3D) scene using OpenGL and C++. The goal was to replicate a 2D image by breaking it down into basic shapes and transforming those into low-polygon 3D models. I chose to create a scene inspired by a simple stack of books with a lamp beside them—something that felt familiar, personal, and achievable within the constraints of the assignment.

The main challenge was taking something from the real world and simplifying it. Think of it like trying to recreate a Lego version of your desk—not every little detail can be captured, but you can still get the point across with the right combination of blocks. That’s exactly how I approached this project. I started with the three books, each represented using box primitives. Boxes are great for blocky shapes like books because they’re efficient and easy to scale and texture.

To add depth and realism to the scene, I textured the books with different colors—red, green, and blue—using royalty-free texture images sized at 1024x1024 pixels, as suggested by the course materials. Texturing helped the books stand out and made them look less like blank shapes and more like actual objects.

For the lamp, I had to combine two primitives—a cylinder for the stand and a cone for the lamp shade. This was one of the more interesting parts of the project because it required stacking and aligning two different shapes to act as a single object. It’s like taking a paper towel roll and putting a party hat on top—it’s a simple idea, but when scaled and positioned properly, it really sells the concept. To complete the look, I textured the pole and shade using two custom textures: one with a fabric-like texture for the head and another greenish texture for the pole. This not only added variety but also allowed me to meet the requirement of applying textures to at least two objects.

Lighting was another important part of this project. I implemented a combination of one directional light (like sunlight) and two point lights (like ceiling lamps) to bring out the shadows and highlights on the objects. I used the Phong lighting model, which simulates how real light interacts with surfaces—kind of like how you can tell whether an object is shiny or dull in real life based on how light bounces off of it. Each object in the scene was assigned a material, which allowed me to control how shiny or matte it appeared, adding to the realism.

Camera control was another area where I invested time. I wanted users to be able to move around the scene using the keyboard and mouse, similar to how players explore in a 3D video game. I implemented WASD keys for basic movement and QE for vertical control, while the mouse allowed users to look around freely. I also added a feature that lets you switch between perspective view (which gives a sense of depth) and orthographic view (which flattens the scene) with the tap of a key. This makes the scene more interactive and allows the user to view it in multiple ways.

One of the things I’m proud of is keeping the project modular. I avoided cramming all of the code into one file or function. Instead, I organized it across classes like SceneManager, ViewManager, and ShaderManager, which made everything easier to debug and maintain. I also created helper functions for reusable actions like setting transformations or applying materials, which made the code cleaner and easier to follow.

Throughout this process, I learned that 3D graphics are not just about complex models or textures—it’s about getting the fundamentals right. Even simple shapes can tell a compelling visual story if they're placed thoughtfully, lit properly, and made interactive. It’s a bit like theater: you don’t always need elaborate props if the lighting, staging, and direction are strong.

In conclusion, this project helped me develop a deeper appreciation for how 3D scenes are built from the ground up. It combined creativity with coding, problem-solving with design. I’m walking away from this assignment not just with a working 3D model, but with a better understanding of how digital worlds are constructed—one triangle at a time.

**References**

* LearnOpenGL. (n.d.). *Lighting*. https://learnopengl.com/Lighting/Basic-Lighting
* FreePBR.com. (n.d.). *Royalty-free textures*. [https://freepbr.com](https://freepbr.com/)
* OpenGL documentation. (n.d.). *GLSL Phong Shading*. https://registry.khronos.org/OpenGL-Refpages/gl4/